Agent-based Multimodal Interface for Dynamically Autonomous Mobile Robots

Donald Sofge, Magdalena Bugajska, William Adams,
Dennis Perzanowski, Alan Schultz
Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375
<sofge | magda | adams | dennisp | schultz>@aic.nrl.navy.mil

Abstract

Agents provide a flexible and scalable method of integrating artificial intelligence techniques on a single cohesive distributed computing system. We have designed and implemented an agent-based interface for autonomous control, and for providing web-based information retrieval, for a dynamically autonomous mobile robot. The robot implements and integrates a variety of artificial intelligence techniques including a multimodal interface that allows natural language understanding, gesture interpretation, simultaneous localization and map-building, object identification and spatial reasoning. The agent-based interface augments these capabilities by providing a method of controlling the robot via the CoABS grid, and by providing the means for the robot/operator to request information available through the grid or through the web.

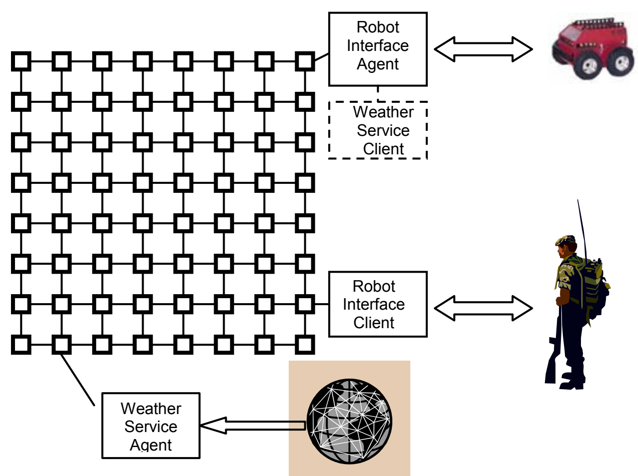

Figure 1 – CoABS Grid Architecture for Dynamically Autonomous Mobile Robots

1. Introduction

The integration of multiple artificial intelligence (AI) techniques onto a real-time system, such as a dynamically autonomous mobile robot, presents a variety of challenges. Such integrated systems often run in a distributed environment. This integration can be enhanced through the use of an agent-based approach to the services offered by each available AI module. Intelligent agents when coupled with grid-based computing provide a compelling paradigm for the design of a distributed AI system. The grid-based computing paradigm allows for a flexible and expandable hardware architecture, while intelligent agents provide services based upon requests submitted via the grid.

In this effort we utilize two agents on the Control of Agent Based Systems (CoABS) Grid in order to demonstrate the effective use of the agent- and grid-based computing paradigm for integrated distributed artificial intelligence systems on autonomous mobile robots (Figure 1).

The remainder of the paper is organized as follows. Section 2 describes our integrated AI architecture for dynamically autonomous robots. Section 3 discusses the need for dynamic autonomy in mobile robots, and how dynamic autonomy is achieved in our system by integrating behaviors such as planning and navigation, prioritization of commands, and interaction with the operator through use of a multimodal interface. In Section 4 we describe the multimodal interface which provides the operator with multiple pathways for communicating commands to the robots, thereby providing a more flexible and robust human-robot interaction and control mechanism. Section 5 describes DARPA's CoABS grid architecture which was used as the backbone for the agents in this effort. Section 6 presents the details of the *Robot Interface Agent, Robot Client,* and *Weather Agent* implemented in this effort to demonstrate the use of agents and the CoABS grid architecture for expanding the capabilities of a highly-integrated distributed AI system, as demonstrated on an autonomous mobile robot platform. Section 7 provides a summary of results and conclusions.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Agent-based Multimodal Interface for Dynamically Autonomous Mobile Robots**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Navy Center for Applied Research in Artificial Intelligence,Naval Research Laboratory,Code 5510,Washington,DC,20375**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
Agents provide a flexible and scalable method of integrating artificial intelligence techniques on a single cohesive distributed computing system. We have designed and implemented an agent-based interface for autonomous control, and for providing web-based information retrieval, for a dynamically autonomous mobile robot. The robot implements and integrates a variety of artificial intelligence techniques including a multimodal interface that allows natural language understanding, gesture interpretation, simultaneous localization and mapbuilding, object identification and spatial reasoning. The agent-based interface augments these capabilities by providing a method of controlling the robot via the CoABS grid, and by providing the means for the robot/operator to request information available through the grid or through the web.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

2. Integrated Goal-Driven Architecture

Our mobile robot architecture is organized around a goal integration and arbitration module as shown in Figure 2. Various capabilities such as speech recognition and natural language understanding, gesture interpretation, and other operator interface modules' outputs are cached, and command prioritization and resolution are performed. The *CoABS Grid Manager* provides a portal for integrating additional capabilities into the architecture. The *CoABS Grid Manager* coordinates all activities over the grid by allowing agents to register and advertise their services, request services from other agents, and transfer information over the grid to fulfill requests.

The *Robot Interface Agent* implements the interface between the grid manager and the *Goal Interpretation and Resolution* module. At present the system includes two CoABS Grid-based services, a *Robot GUI Agent* which implements a local or remote screen-driven interface to the robot, and a *Weather Service Agent* which provides a connection between the robot and the *World Wide Web* in order to access real-time weather service information.

The *Robot Interface Agent* and *CoABS Grid Manager* enhance the scalability of the architecture by providing a means to add new capabilities easily either directly through the use of other CoABS Grid agents, or through software agents and services accessible via the *World Wide Web*.

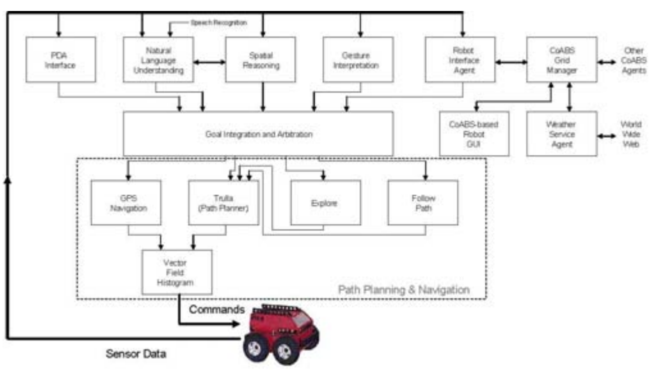

Figure 2 – Integrated Goal-Driven Architecture

Once goals are interpreted and resolved, they are passed to the *Path Planning and Navigation* routines. This section integrates low-level behaviors such as obstacle avoidance, exploration and path planning using the Vector Field Histogram (VFH) method [1]. Both short-term and long-term maps are maintained. The maps (not shown in Figure 2) also are important for several of the other processes such as *Spatial Reasoning*, *PDA Interface*, and *Robot GUI*.

2.1 Evidence Grid Representation

A key to achieving a robust yet scalable architecture for autonomous mobile robots is the use of a common representation for integrating motion planning and navigation. The central unifying representation used in our work is the evidence grid [2].

An evidence grid is a probabilistic representation of Cartesian space which divides the space of the robot into a grid of cells. The probability of an individual cell being occupied is given by a real-valued number in the range (-1, 1), with (1) indicating that the cell is occupied and (-1) indicating that it is unoccupied. In our system the evidence grid is populated based upon the returns from the robot's sensors, including sonar sensors and a planar structured-light sensor. Whereas the sonar sensors are better at providing evidence that an area is empty, the structured light sensor is better at providing evidence that an area is occupied due to its planar 2D nature (it may miss 3D objects above or below its plane of sensitivity, therefore such areas cannot be ruled empty). These observations are accumulated and the evidence grid is updated using a Bayesian update rule [2]. The grid information is maintained in short-term and long-terms maps. The short-term map shows what the robot senses in its immediate spatial environment, whereas the long-term map is built up over time, and would for example be use to show the layout of a room or series of rooms. The short-term map is used to update the long-term map. Our use of the evidence grid as a common representation is described in detail in [7].

3. Dynamic Autonomy in Mobile Robots

Effective interaction with a mobile robot requires that the robot be capable of acting or responding at a level of autonomy appropriate to the task at hand [6]. Given a suitable interface, the human operator is able to interact with the robot in a more human-centric manner by providing verbal commands and gestures to the robot to delegate tasks in a manner appropriate to the task. While some circumstances may require a very fine-grained level of control by the operator, other tasks may be specified less precisely. The use of dynamic autonomy in mobile robots makes the robots more versatile and provides a more flexible and operator-friendly interface.

These multiple pathways for directing the robot support dynamic autonomy by extending the operator's capabilities to utilize the robot in a more cooperative or collaborative mode, perhaps working along side the robot performing a task rather than focused solely upon controlling what the robot is doing.

In our system we provide dynamic autonomy through a variety of behaviors the robot may execute. These include collision-free navigation, path following, multimodal interfaces (e.g. "Robot, go over there" *with accompanying synthetic or natural gesture*), automatic prioritization of multiple command directives (e.g. "Go over there", then "Come here"), and feedback from the robot to the operator via voice synthesis (using ViaVoice) and text strings (to the computer) requesting clarification if the robot isn't able to resolve conflicts and ambiguities to understand the command(s). The operation of the natural language interface and its integration with the gesture interpretation process and other command input modalities is discussed in greater detail in the next section.

4. Multimodal Interface for Mobile Robots

A key aspect of our approach to development of dynamically autonomous mobile robots is the use of a multimodal interface for interacting with the robot [3]. Multiple modes of communication provide the operator with a natural and more efficient interface to the system. It also provides redundancy in case of subsystem failures. This redundancy can be effectively used to resolve ambiguity, such as when a command is communicated in more than one way to the robot (e.g. verbally and through a gesture).

In our system commands may be communicated to the robot through spoken language, through a simple GUI on a Personal Digital Assistant (PDA) interface (connected via a wireless network) or an enhanced GUI on a PC, or through hand and arm gestures [4]. The robot passes back a variety of information such as sensor readings, video, navigation maps built using its array of on-board sensors [7], and the status of commands sent to it (e.g. acknowledgement of receipt and completion). This information is then passed along to the operator based upon the various device types in use. For example, video streams will be sent to the desktop GUI but not to the PDA, whereas map data will be sent to both. Some types of feedback, such as requests for clarification or goals completion statements, will be spoken by the robot through its on-board voice synthesizer, and also sent as a text string back to the desktop GUI.

4.1. Gesture Recognition Process

The gesture recognition process utilizes a structured-light rangefinder which emits a horizontal plane of laser light. A camera mounted on the robot just above the laser is fitted with an optical filter which is tuned to the frequency of the laser. The camera registers the reflection of the laser light off of objects in the room and generates a depth map (XY) based upon location and pixel intensity. The data points for bright pixels (indicating closeness to the robot) are clustered, with an average taken, and in particular if a cluster is significantly closer to the robot than average objects, it is interpreted as being a hand. Hands locations are stored from several consecutive frames, and the positions of the hands are used to generate trajectories for the gesture command. The trajectories are analyzed to determine if they represent valid gesture commands, and if so which commands they match. The gesture is then queued so that the multimodal interface, upon receiving another command (e.g. verbal - "Go over there."), can retrieve the gesture from the gesture queue and combine it with the verbal command in the command interpretation system.

4.2. Natural Language and Spatial Reasoning

A natural language interface is used which combines a ViaVoice front-end with an in-house developed deep parsing system [6]. This gives the robot the capability to parse utterances, providing both syntactic representations and semantic interpretations. The semantic interpretation subsystem is integrated with the other sensor and command inputs through use of a command interpretation system. The semantic interpretation, interpreted gestures from the vision and/or light-striping sensors, and command inputs from the computer or PDA interface are compared, matched and resolved in the command interpretation system.

Building upon the existing framework of natural language understanding with semantic interpretation, and utilizing the on-board sensors for detecting objects and map-building through use of evidence grids, we are developing a spatial reasoning capability on the robot [8]. Spatial reasoning is important not only for solving complex navigation tasks, but also because we as human operators often think in terms of the relative spatial positions of objects. For example, we may want to give the robot a command such as "Go behind that building and then proceed West 20 feet. Then patrol until further notice." Or, in an office or laboratory setting, "Go between the desk and the chair, out the door, and down the hall to the left." Spatial reasoning increases the dynamic autonomy of the system by giving the operator a less restrictive vernacular for commanding the robot.

Another benefit of incorporating a spatial reasoning component is that the robot can provide feedback to the operator in spatial terms. For example, the following is a typical dialogue between the robot and operator.

Human: "How many objects do you see?"
Robot: "I see 5 objects."
Human: "Where are they located?"

Robot: "Two are behind me to my left, one is directly in front of me, and one is in front of me to my right."

If the human operator labels the objects described, or if an object identification agent is added to the system (e.g. for recognizing tank-like objects), we can imagine the following dialogue.

Human: "Go behind that building and report what you see."
Robot: "OK. I'm on my way."
(*robot navigates to goal avoiding obstacles*)
Robot: "I made it to the goal. "
"I see 2 tank-like objects, and four unrecognized objects."
Human: "Where are the tank-like objects located?"
Robot: "The first tank-like object is 25 feet in front of me and to my right.
The second tank-like object is 75 feet in front of me and just to the left.

Establishing a spatial language [8] is necessary so that it is clear what is meant by spatial references generated both by the human operator as well as by the robot. Thus, if the human commands the robot, "Turn left," the robot must know whether the operator refers to the robot's left, or the operator's left. In the dialog above, the robot places the second object "just to the left of the first object." Does the robot mean its left, or the first tank's left? We are currently investigating this through use of human-factors experiments where individuals who do not know the spatial reasoning capabilities and limitations of the robot provide instructions to the robot for performing various tasks where spatial referencing is required. The results of this study will be used to enhance the multimodal interface by establishing a common language for spatial referencing which incorporates those constructs and utterances most frequently used by untrained operators for commanding the robot.

The primary goals of this effort were to demonstrate the use of agent technology to enhance a highly-integrated distributed AI system implemented on an autonomous mobile robot platform, and to demonstrate the use of the agents and integrated AI capabilities in real-time. This required implementation of a GUI as a CoABS Grid-based interface agent such that a client located anywhere on the grid could access and control the robot. Information (including video) would flow in real-time from the robot through the *Robot Interface Agent* to the *Robot Client*, and commands would flow from the *Robot Client* through the *Robot Interface Agent* to the robot. A secondary goal was to demonstrate that the robot (or robot operator through the robot) could access other grid-based services, including services which glean information from the web.

The next section will describe the CoABS Grid and provide a brief overview of the CoABS project, the need for the Grid by the U.S. military and the capabilities offered by the CoABS Grid for agent-based systems, such as the integrated AI system deployed on an autonomous mobile robot. The following section (Section 6) will detail our efforts in applying the grid technology to our system, and describe the agents developed to demonstrate the use of the CoABS Grid in a real-time dynamically autonomous mobile robot application.

5. CoABS Grid

The CoABS Grid emerged from the DARPA IPTO program Control of Agent Based Systems under which this effort was funded. CoABS is a Department of Defense program designed to foster development and encourage the use of agent-based systems to improve military command, control, communications and intelligence gathering (C^3I). The primary emphasis in CoABS has been the development of a prototype middleware, the CoABS Grid, for coordinating and managing large numbers of cooperating agents over a heterogeneous, loosely coupled network. The CoABS Grid integrates heterogeneous agents, object-based applications, and legacy applications into a common framework whereby agents can register their services dynamically, advertise their capabilities, search for needed services or capabilities, and transmit and receive messages between agents.

The grid also provides a variety of services to facilitate its use by agents. These include a logging service to log both message traffic and other information, a security service (to provide authentication, encryption, and secure communication), and an event notification service to allow agents to register, deregister, or change their advertised attributes. These services are handled by the Grid Manager, a utility for monitoring interactions over the grid (Figure 3).

The grid is important to the military for potential use with mobile autonomous robotic systems because it provides a logical, highly structured backbone for coordinating these assets and performing battlefield C^3I. This work is one of the first to successfully demonstrate a real-time grid-enabled dynamically autonomous mobile robot which can be supervised from a grid-enabled client located anywhere on the grid, and feed back information from the robot (including video) to the remote operator. It can also be controlled by a local operator via a PDA, voice commands or gestures.

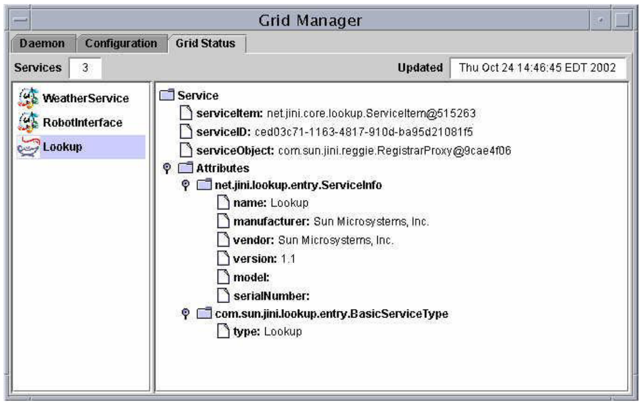

Figure 3 – CoABS Grid Manager GUI

The *Robot Interface Client GUI* can be dynamically configured to control a variety of different robots by selecting them by name. However, the user can only interact with the single currently selected robot. We are working on scaling the system up to direct multiple robots simultaneously (at a team level) from a single interface [5]. Also, we are planning to add an arbitration mechanism to address the issue of a robot receiving multiple (possibly conflicting) commands simultaneously from multiple operators.

6. Agents for an Autonomous Mobile Robot

First, a *Robot Interface Agent* is registered with the Grid Manager and advertises its services for providing an interface to the robots. An operator located somewhere on the grid may then start a *Robot Interface Client* which identifies and begins a dialogue with the *Robot Interface Agent*, which then creates a local GUI for the operator to interact with the robots. Once the interface is available, the operator will begin receiving information from the robot including video feed, status information, etc. The operator, through the client, may also begin issuing commands for the robot to perform certain actions. The *Robot Interface Client GUI* is shown in Figure 4.

A *Weather Service Agent* is registered and advertises its services for providing real-time weather forecasts for cities distributed around the world. The weather agent retrieves and parses html from web pages provided by the National Weather Service. The *Robot Interface Agent* starts a *Weather Service Client* so that the operator can retrieve weather forecasts through the robot interface. We currently use the speech recognition and natural language parsing modules to let an operator make a verbal request for weather information to the robot. The robot "understands" that request and submits the weather request through the *Robot Interface Agent* to the *Weather Service Agent*. The *Weather Service Agent*, which has a web connection, retrieves the appropriate web page and parses it to extract the desired weather report. The weather report is sent through the *Robot Interface Agent* to the *Robot Interface Client* (where it is displayed on the GUI of the client computer) and is also sent to the robot, where it is read back to the operator using the on-board voice synthesizer.

The GUI (Figure 4) is used to control the robot as well as present the current status of the robot and other information useful to the operator, such as weather reports. The message window in the upper left corner (label 1) displays the weather reports and status information on tasks assigned to the robot, such as when each task has been completed. The messages are color-coded according to the priority level of the message.

The robot selector window to the right of the message window (label 2) shows the robot currently being controlled. The operator can switch between robots using the pull-down menu. The window below the robot selector window (label 3) shows meteorological data collected by the robot, currently wind speed and direction.

The weather request window (label 4) allows the operator to select a city via a pull-down menu and then by clicking <Submit> to send a request to the *Weather Service Agent* for a weather report. As discussed previously the operator may also get the weather report by a making verbal request to the robot. Once retrieved the weather report is displayed in the message window and it is also spoken by the robot.

The map window (label 5) is used to show the robot's current map of its environment. Maps may be built up through exploration of the environment, stored, and later reloaded. Thus, the robot may begin navigating in a previously explored area by loading a pre-stored map of that area, and then augmenting that map using its on-board sensors. The map window also serves as a command input mechanism for the operator to direct the robot to a specific location on the map, or to trace a path on the map for the robot to follow. The operator may use a mouse (or touch screen on the PDA) to indicate the goal point or path on the map. A trace of the recent path of the robot can be overlayed onto the map to provide additional information.

The large window to the right of the map (label 6) shows an aerial (e.g. satellite) image of an area, and is used in conjunction with the robot's on-board GPS navigation system for outdoor navigation. The operator can pan and zoom the overhead image, and as with the map window the operator can direct the robot to a specific location by clicking on the map or have it follow a path by tracing it on the map. As in the map window, a trace of the recent path of the robot can be overlayed onto the image map.

In the bottom left corner of the GUI (label 7) the video window shows the current video feed from the robot. This provides the operator with a remote view of what the robot is "seeing", with a video frame refresh rate sufficient for teleoperation of the robot using the joystick interface. Work currently underway will in the future allow us to direct the robot by clicking on the video image.

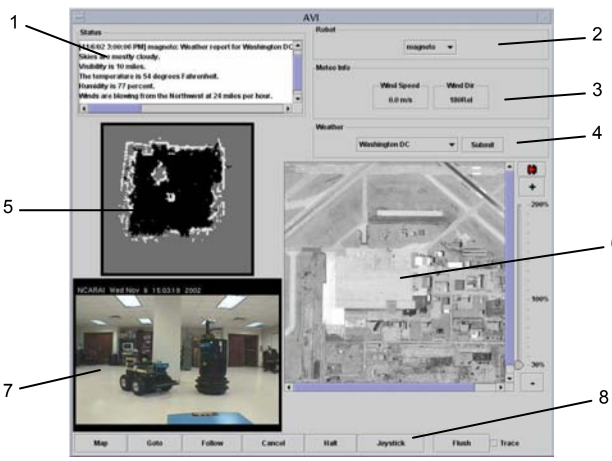

Figure 4 – Robot Interface Client GUI

The joystick interface (not shown in Figure 4) is another window which pops up if the <Joystick> button (label 8) is pressed. This provides yet another means for controlling the robot. The operator either clicks or holds the position of a virtual joystick in order to move the robot forward (up), backward (down), turn left (left), turn right (right), or some combination of the above.

7. Summary and Conclusions

The agent-based architecture provides a natural and highly scalable approach to integrating new AI capabilities into the system. We are expanding the capabilities of our robots in a variety of ways, including adding spatial reasoning to allow the robots to sense and make decisions with regard to objects in their immediate environment, adding a cognitive architecture to allow the robots to reason in a more human-like manner, and enhancing the multimodal interface to allow a single operator to control multiple robots simultaneously (robot teams).

Additional agents could be implemented which take advantage of the grid architecture, both enhancing the capabilities of the robot directly, and through providing access to the operator in the field for services through the robot (such as the weather). The *Weather Service Agent* demonstrates that a grid-enabled robot can utilize an external agent which accesses the web in order to retrieve potentially important information.

Acknowledgements

Support for this effort was provided by the Defense Advanced Research Projects Agency IPTO CoABS Program under MIPR 02-M733-00.

References

[1] Borenstein, J. and Koren, Y. (1991). "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions on Robots and Automation*, IEEE: New York, 7(3):278-288.

[2] Moravec, H. and Elfes, A. (1985). "High Resolution Maps from Wide Angle Sonar," In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 116-121.

[3] Perzanowski, D., Adams, W., Schultz, A., and Marsh, E. (2000). "Towards Seamless Integration in a Multimodal Interface," In *Proceedings 2000 Workshop Interactive Robotics and Entertainment*, pages 3-9, Menlo Park, CA.

[4] Perzanowski, D., Schultz, A.C., and Adams, W. (1998). "Integrating Natural Language and Gesture in a Robotics Domain," In *Proceedings IEEE International Symposium on Intelligent Control*, pages 247-252, Piscataway, NJ.

[5] Perzanowski, D., Schultz, A.C., and Adams, W., Bugajska, M., Marsh, E., Trafton, J.G., Brock, D., Skubic, M., and Abramson, M. (2002). "Communicating with Teams of Cooperative Robots," In (Shultz, A.C. and Parker, L.E., Eds.), *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings of the 2002 NRL Workshop on Multi-Robot Systems*, Kluwer.

[6] Perzanowski, D., Schultz, A., Adams, W., and Marsh, E. (1999). "Goal Tracking in a Natural Language Interface: Towards Achieving Adjustable Autonomy," In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 208-213, Monterey, CA.

[7] Schultz, A., Adams, W., and Yamauchi, B., (1999). "Integrating Exploration, Localization, Navigation and Planning Through a Common Representation," *Autonomous Robots*, 6(3), Kluwer.

[8] Skubic, M., Perzanowski, D., Schultz, A., and Adams, W. (2002). "Using Spatial Language in a Human-Robot Dialog*," Proceedings 2002 IEEE Conference on Robotics and Automation*, IEEE.